

cytometree: a binary tree algorithm for automatic gating in cytometry analysis

May 31, 2018

Daniel Commenges^{1,2*}, Chariff Alkhassim^{1,2}, Raphael Gottardo³, Boris Hejblum^{1,2}
and Rodolphe Thiébaud^{1,2,*}

¹ Inserm, Bordeaux Population Health Research Center, UMR 1219, INRIA
SISTM, Univ. Bordeaux, ISPED, F-33000 Bordeaux, France.

² Vaccine Research Institute VRI, Groupe Henri-Mondor Albert-Chenevier,
F-94010 Creteil, France.

³ Vaccine and Infectious Disease Division, Fred Hutchinson Cancer Research
Center, 1100 Fairview Ave N, Seattle, WA, 98109, USA.

*To whom correspondence should be addressed.

Abstract

Motivation: Flow cytometry is a powerful technology that allows the high-throughput quantification of dozens of surface and intracellular proteins at the single-cell level. It has become the most widely used technology for immunophenotyping of cells over the past three decades. Due to the increasing complexity of cytometry experiments (more cells and more markers), traditional manual flow cytometry data analysis has become untenable due to its subjectivity and time-consuming nature.

Results: We present a new unsupervised algorithm called “**cytometree**” to perform automated population discovery (aka gating) in flow cytometry. **cytometree** is based on the construction of a binary tree, the nodes of which are subpopulations of cells. At each node, the marker distributions are modeled by mixtures of normal distribution. Node splitting is done according to a normalized difference of Akaike information criteria (AIC) between the two models. Post-processing of the tree structure and derived populations allows us to complete the annotation of the derived populations. The algorithm is shown to perform better than the state-of-the-art unsupervised algorithms previously proposed on panels introduced by the Flow Cytometry: Critical Assessment of Population Identification Methods (FlowCAP I) project. The algorithm is also applied to a T-cell panel proposed by the Human Immunology Project Consortium (HIPC) program; it also outperforms the best unsupervised open-source available algorithm while requiring the shortest computation time.

Availability: An R package named “**cytometree**” is available on the CRAN repository.

Contact: daniel.commenges@u-bordeaux.fr; rodolphe.thiebaut@u-bordeaux.fr

Supplementary information: Supplementary data are available.

Keywords: Flow cytometry, automated gating, binary tree, mixture of distributions

1 Introduction

Recent technological advancements in instrumentation have transformed the field of flow cytometry by enabling rapid, multidimensional quantification of millions of individual cells to define cellular subpopulations and assess cellular heterogeneity (O'Neill *et al.*, 2013; Aghaeepour *et al.*, 2013). Traditionally, flow cytometry data are analyzed manually by drawing geometric shapes (referred to as ‘gates’) around populations of interest in a series of 1-2 dimensional data representations. This process, known as gating, is time-consuming and highly subjective (Aghaeepour *et al.*, 2013). Modern instruments including both flow and mass cytometers are now capable to quantify between 20 and 50 proteins, leading to high-dimensional data that are impossible to exhaustively explore by manual analysis. Several supervised and unsupervised algorithms have been proposed for automatic gating, including model-based clustering approaches (Lo *et al.*, 2008; Finak *et al.*, 2009; Chan *et al.*, 2008; Pyne *et al.*, 2009; Qian *et al.*, 2010; Aghaeepour *et al.*, 2011; Cron *et al.*, 2013a), a nonparametric density-based approach (Naumann and Wand, 2009), and a spectral clustering algorithm (Zare *et al.*, 2009), among others. A number of these have been compared in the open competition set-up by the FlowCAP consortium (Aghaeepour *et al.*, 2013) and many developments have followed as a result (Brinkman *et al.*, 2015, 2016; Finak *et al.*, 2014) as reviewed in Saeys *et al.* (2016). Automated cell classification has also been applied to mass cytometry by time of flight (CyTOF) data (Lee *et al.*, 2017). Many of these algorithms performed rather well on the FlowCAP benchmark data. However, no single method was uniformly superior on all datasets. Additionally, some of these methods were very computationally demanding and no method led to biologically interpretable cell populations (i.e. population labels are exchangeable). To overcome these problems, supervised algorithms including flowDensity (Malek *et al.*, 2015) and OpenCyto (Finak *et al.*, 2014) have been proposed and compared to manual gating for several panels of cells analyzed by nine laboratories by the Human Immuno Phenotyping Consortium (HIPC). However, being supervised, these approaches require significant tuning and restrict the exploration of flow cytometry data to pre-specified cell populations.

Here, we propose a new method that is fast, compares favorably to state-of-the-art approaches, and leads to biologically interpretable populations. It uses the same basic idea that experimentalists utilize when analyzing data: a given cell either expresses or does not express a given protein (i.e. the marginal distribution of each marker is mostly bimodal). That is, for most markers, the cells will be either negative (-) or positive (+). We approximate the distribution of each marker by a mixture of two normal distributions. This process allows us to cycle through all markers, and to build a binary tree, the leaves of which are the terminal subpopulations. The annotation is completed using a post-processing procedure. We call this new method “**cytometree**”.

Our paper is organized as follows: We first present the **cytometree** algorithm, then an illustration of the outputs of the program using the HIPC T-cell panel. Finally, we demonstrate its utility and performance on FlowCAP I and FlowCAP III challenge benchmark data.

2 Methods

2.1 Principle

The distribution of each marker is modeled as a mixture of two normal distributions. A criterion, noted D , is computed for each marker to measure the improvement brought by the mixture over a single normal. The marker with the highest value of the criterion is chosen to be the first node of the tree. The population of cells is divided into two subsets, provisionally annotated negative (-) and positive (+) for this marker. The same procedure is then applied for each subpopulation, and thus a binary tree is obtained. The tree growth is stopped when the highest value of the criterion is below a pre-specified threshold. The leaves of the tree are the final subpopulations obtained by this algorithm. Branches of the tree (i.e. the gating path) lead to a biologically interpretable annotation (e.g. CD3+/CD4+) for each subpopulation given the markers that were used at each node in the path (from the root to the given leaf). However, a given path may not make use of all available markers, as some markers might not exhibit bimodality, or have a D value always lower than other markers in competition. As such, we also propose a post-processing annotation algorithm to generate alternative population labels that make use of all markers.

2.2 Binary tree algorithm based on difference of normalized AIC

In this section, we describe more precisely the construction of the binary tree. Let us denote by s_j the set of available markers at node j , that is, the set of markers that have not been used for defining parent nodes of j , and by n_j the number of cells at node j . At a given node j , **cytometree** inspects whether the population of cells can be split by exhaustively searching over all available markers in the set s_j , the marker for which the fluorescence distribution of the observed cells maximizes the criterion D , defined as

$$D = \frac{\text{AIC}_1 - \text{AIC}_2}{n_j},$$

where AIC_1 and AIC_2 are the conventional Akaike criterion values for the one and two component mixture models, respectively. D is a normalized version of the difference of Akaike criterion (AIC) (Commenges *et al.*, 2008). The advantage of this criterion is that it does not depend on n_j , but estimates the difference of Kullback-Leibler divergences from the true distribution (the true distribution is the distribution from which the data are supposed to be generated). A difference of 0.1 was considered “large”. Thus, for each marker $m \in s_j$, the distribution of the fluorescence intensity $\mathbf{Y}_m^{(j)}$ can be modeled as either a single normal with mean μ and variance σ^2 or a two component mixture model as follows:

$$f_{Y_m^{(j)}}(y) = \pi\phi(y; \mu_1, \sigma_1^2) + (1 - \pi)\phi(y; \mu_2, \sigma_2^2), \quad (1)$$

where $\phi(\cdot; \mu, \sigma^2)$ is the normal density of mean μ and variance σ^2 . The parameters of the mixture $(\pi, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$ are estimated by maximum likelihood using an EM algorithm (Fraley and Raftery, 2002; Fraley *et al.*, 2012). Then, the criterion for marker m , D_m , is computed using the likelihoods of the one and two component mixture models. The maximum value of D over all markers is defined as $D^* = \max_m D_m$. If D^* is above a pre-specified threshold t^* , the population at node j is split according to the values of the marker m^* that achieved this maximum and two child nodes are obtained. Cells with $y_m^j < y_{\min}$ (resp. $y_m^j > y_{\min}$) form the subpopulation of the left (resp. right) child, where $y_{\min} = \operatorname{argmin}_{y \in (\hat{\mu}_1, \hat{\mu}_2)} f_{Y_m^{(j)}}(y)$. If $D^* < t^*$, the tree growth is stopped, and j is a leaf of the tree. The threshold t^* can be adjusted to find more or fewer populations; our experience showed that values between 0.1 and 0.2 generally give good results.

When the tree has stopped growing, the leaves yield a partition of the data into L subpopulations $\mathcal{P} = \{S_1, \dots, S_L\}$.

It can be shown that the algorithm runs in linearithmic fashion as a function of the number of cells n , that is, the complexity is in $\mathcal{O}(n \log n)$ (see Figure S6 in Supplemental material for an empirical check). The computational cost increases linearly with the dimension (number of markers) for each node. Moreover, the computational cost is linear in the number of nodes and the number of nodes is lower than twice the number of leaves, that is the sub-populations. The number of leaves of the tree is likely to increase slightly with number of dimensions but the number of sub-populations cannot be very high. Assuming the number of sub-population is bounded, the complexity is essentially linear in the dimension.

2.3 Annotation algorithm

Given that the binary tree construction is unsupervised and depends on a pre-specified threshold t^* , some of the available markers may not have been used to find the different cell subpopulations (i.e. some markers in some paths never pass the threshold, or always have D values lower than other markers). To recover a complete annotation using all available markers, we devised a post-processing exhaustive annotation method. This step can be supervised or unsupervised. In the supervised option, the number of expression levels of each marker is fixed by the user, in the unsupervised option it is proposed by the algorithm based on the D criterion.

Supervised option. In the supervised option, we wish to annotate the subpopulations for some or all of the markers. In general, we wish to annotate the subpopulations as positive or negative for the chosen markers. For each chosen marker, we rank the means of the L subpopulations, forming $(S_{(1)}, \dots, S_{(L)})$. Then we form $L - 1$ partitions of the L subpopulations into two groups: the partition p , for $p = 1, \dots, L - 1$, is (G_p^1, G_p^2) , with $G_p^1 = \cup_{j=1,p} S_{(j)}$ and $G_p^2 = \cup_{j=p+1,L} S_{(j)}$. Then we find the best partition G_{p^*} in the sense of minimizing the within-cluster

variance:

$$p^* = \operatorname{argmin}_p \sum_{k=1}^2 |G_p^{(k)}| \operatorname{var} G_p^{(k)}, \quad (2)$$

where $|G_p^{(k)}|$ is the cardinal of $G_p^{(k)}$. This is the same formula that is used in the K-means algorithm, but only $L - 1$ partitions are tried and the observations are one-dimensional (since we work marker-by-marker). Thus, this step of the algorithm is very fast. Finally, we label subpopulations $(S_{(1)}, \dots, S_{(p^*)})$ as negative “(−)” and $(S_{(p^*)}, \dots, S_{(L)})$ as positive “(+)” for the marker at hand.

We can perform the same type of algorithm for partitioning the subpopulations into three groups, “−”, “+” and “++” for some markers. Here the number of partitions is $(L - 1)(L - 2)/2$. This can also be done for the markers used in the tree. As an example, we may wish to find three levels of CD45RA; this is exemplified in HIPC Patient 12828 replicate 3 from the NHLBI dataset; see Supplementary Material S1.

Unsupervised option. In this option, for the markers not used in the tree, we compute the D criterion comparing the fits of the marginal distribution obtained by one normal distribution and by a mixture of two or three normal distributions for judging whether there are one, two, or three groups. For the markers used in the tree, we compute the D criterion to compare the fits obtained by a mixture of two and three normal distributions.

2.4 F -measure

The F -measure is a popular metric to evaluate clustering methods. It can be used as a way to summarize the concordance between two classification methods (one being set as the reference). This measure is the harmonic mean of precision and recall (Aghaeepour *et al.*, 2013). The precision is the number of cells correctly assigned to a given cluster divided by the total number of cells assigned to this cluster. The recall is the number of cells correctly assigned to a given cluster divided by the number of cells that should be assigned to this cluster according to the reference method. The total F -measure is then calculated for each combination of the reference clusters and the predicted clusters. It yields a value of $[0, 1]$, with 1 indicating a perfect match between the two clustering methods.

2.5 Benchmarking

2.5.1 FlowCAP I challenge

Several unsupervised algorithms have been compared to manual gating done by a consensus of 8 manual operators (from 8 different laboratories) on 5 data sets. These data sets included four human data sets: graft-versus-host disease (GvHD), diffuse large B-cell lymphoma (DLBCL), symptomatic West Nile virus (WNV), and normal donors (ND); the fifth was a mouse data set (hematopoietic stem cell transplant (HSCT)). Each of the 5 data sets includes multiple samples

of up to 10^5 cells measured on a maximum of 10 markers. The results were set to be used as benchmark data in the FlowCAP I challenge (Aghaeepour *et al.*, 2013). The data were downloaded from the FlowCAP project website as part of the FlowCAP I challenge.

2.5.2 HIPC T-cell panel

The Human Immune Phenotyping Consortium (HIPC) was developed with the aim of standardizing flow cytometry immunophenotyping in clinical studies. Finak *et al.* (2016) investigated whether automated gating could help standardizing flow cytometry data analysis. We used a part of the data collected in this study to assess the performance of the `cytometree` algorithm, focusing on the T-cell panel. Seven laboratories (or centers) stained three replicates of three cryopreserved PBMC samples and returned usable FCS files to the main center for manual and automated gating. The automated gating used a combination of algorithms including `flowDensity`, which is a supervised algorithm. Data sets are publicly available from the ImmuneSpace database (Brusic *et al.*, 2014) and were used as part of the FlowCAP III challenge.

We reproduced the variability analysis of the estimated proportions p_{rij} in replicate r of sample i in center j , of each subpopulation of cells presented in Finak *et al.* (2016). Denoting by $Y_{rij} = \log p_{rij}/(1 - p_{rij})$, the logit of these proportions, the model was as follows:

$$Y_{rij} = \mu + \alpha_i + \beta_j + \varepsilon_{rij}, \quad (3)$$

where μ is the intercept, α_i is a sample random effect, β_j is a center random effect, and ε_{rij} is the residual error. All these random variables are assumed to be independent normal with zero means and with variances σ_α^2 , σ_β^2 and σ_ε^2 , respectively. This model allows us to estimate and quantify the different sources of variability. There is one difficulty in this analysis in that the logit of zero is infinite. For this reason, as in Finak *et al.* (2016), we excluded zero values.

3 Results

3.1 Example of a T-cell sample analysis

For the purpose of illustrating how the algorithm works and what the output looks like, we show some results for a single T-cell sample: the Stanford FCS data for sample 1349, replicate 3 from the HIPC dataset (Finak *et al.*, 2016). The fits for the single normal and the mixture of two normals are computed for all markers, and the differences of normalized AIC (D values) are computed. Figure 1 shows the fits obtained with the mixture compared to non-parametric fits obtained by a kernel method; note the very good fit obtained for the CD4 by the mixture of two normal distributions. The CD4 had the best D criterion (1.31); the first node is declared to be CD4 and is labeled “CD4.1”. Cells are then separated into provisionally negative or positive CD4 groups. Again, the values of the D criterion are computed for all markers except CD4 in the two populations; in both cases, CCR7 wins. Thus, two nodes are created, “CCR7.2” and “CCR7.3”; the fits of the mixture are shown for these two distributions. The tree growth continues until the maximum D criterion value is smaller than 0.1. The tree obtained is displayed in Figure 2.

Although **cytometree** approximates univariate distributions by mixtures of normal distributions, it is possible to reconstitute a bivariate scatter plot such as that analyzed visually in manual gating. The scatter plots obtained for CD45RA and CCR7 for CD4+ cells by manual gating and **cytometree** are displayed in Figure 3, which shows that the two are almost identical for two patients; however for patient 12828a, **cytometree** fails to split the CCR7 population in CD45RA+ and CD45RA- due to the very small number of cells that may constitute this subpopulation.

The next step of the algorithm is the annotation process. This is necessary because although the binary tree gives an annotation for each explored subpopulation, the annotation remains incomplete as some markers may be left unused in the tree growth process. So for all markers, we apply the annotation algorithm described in Section 2.3 to gather the found subpopulations in two or three groups. Results for patient 1349, replicate 3 are displayed in Figure 4, which shows the distribution of the markers for all the subpopulations found, and the result of the clustering algorithm. The subpopulations found in the tree are thus annotated, and a table is constructed to describe them and give the proportion of each subpopulation in the sample. For the chosen sample, the results are shown in Table 1, together with the proportions that were found for the same sample by manual gating in the Stanford center.

3.2 cytometree obtains the best results in FlowCAP I

Table 2 shows the performance of **cytometree** on the FlowCAP I data, compared to the four best performing methods reviewed by Aghaeepour *et al.* (2013) : ADICyt (Chan *et al.*, 2008), flowMeans (Aghaeepour *et al.*, 2011), FLOCK (Qian *et al.*, 2010) and FLAME (Pyne *et al.*, 2009). The F -measures were computed

for all samples available for a given dataset and the mean over all samples is reported. The best open source unsupervised algorithm in the FlowCAP I study appears to be flowMeans. `cytometree` nearly always obtains the highest values, and the mean F -measure is 0.90 for the default value $t^* = 0.1$, making it the best unsupervised approach in the completely automated challenge. We explored a range of values of t^* between 0.05 and 0.25; the F -measure is rather stable on this range. In terms of computing time `cytometree` was one of the fastest algorithms, even faster than flowMeans, taking on average about one minute per sample.

3.3 HIPC T-cell panel

3.3.1 `cytometree` obtains high F -measures on the HIPC T-cell panel

We first compared the F -measures obtained by `cytometree` and flowMeans for the nine sample files (three replicates for three samples) taking as reference the manual gating of the seven centers. The results displayed in Figure 5 show that in most cases, the F -measures obtained by `cytometree` were better than those obtained by flowMeans.

3.3.2 Estimation of proportions of subpopulations and their variabilities compared to manual gating

One of the goals of the method is to find proportions of pre-specified subpopulations of cells. Often the algorithm finds more subpopulations than the pre-specified ones. It is generally possible to group the finer partition that has been found to find the proportions of pre-specified populations; an example is given in Table 1. However, the algorithm has difficulties in some samples to find subpopulations representing less than 1% of the data. This is especially the case for activated T-cells, for which the number of cells can be less than 1%. For this population, the variability was larger than that of the central gating.

We performed the variability analysis based on model 3. Figure 6 displays the center, biological, and residual variabilities for `cytometree` and the manual gating method for six subpopulations of the HIPC T-cell panel. The variability of `cytometree` was similar to that of manual gating, except for CD8 effector T cells. This is in line with the results presented in (Finak *et al.*, 2016), where the authors showed that the CD8 effector T-cell subset was problematic due to poor separation between the HLA-DR- and HLA-DR+ populations.

3.3.3 Discovery of new populations

As discussed previously, `cytometree` is unsupervised and defines cell sub-populations by exhaustive 1-dimensional thresholding of all markers. For example, the T-cell subpopulation labeled 9 in Figure 4 is expressing both CD4 and CD8. Although this population was likely not of primary interest in the manual analysis, `cytometree` builds sample-specific tree patterns and is able to detect such a population, which has been described in several pathological conditions as well

as in normal individuals (Zuckermann, 1999; Parel and Chizzolini, 2004; Quandt *et al.*, 2014).

4 Discussion

cytometree is an unsupervised algorithm for flow cytometry that exhibits better performance in terms of the F -measure than the best unsupervised algorithms, as tested on FlowCAP I data and on the HIPC T-cell panel (FlowCAP III). High F -measure values have been reported by Li *et al.* (2017) who proposed a deep learning algorithm, DeepCyTOF; these values, however, are not comparable to those of unsupervised algorithms; in a very recent paper, Lux *et al.* (2018) report rather modest F -measure values for DeepCyTOF. Other algorithms (Anchang *et al.*, 2014; Samusik *et al.*, 2016) have used binary trees, but as a secondary step; **cytometree** directly starts building the tree (see Supplemental material Sections 4 and 5).

One feature of the algorithm is its numerical simplicity and stability. In particular, mixtures of normal rather than skewed t -distributions (Pyne *et al.*, 2009) were used; in spite of (or thanks to) this simplicity, **cytometree** obtains better F -measures than methods using skewed t -distributions for three reasons: (i) it is simpler and thus more stable; (ii) most of the distributions are not very skewed; and (iii) for moderately skewed distributions the cut-off points obtained with normal and skewed t -distributions are not very different. We show in Figure S2 of the Supplementary Material the results obtained with Flame and **cytometree** for one of the most skewed distribution that we have found in the diffuse large B-cell lymphoma (DLBCL) dataset; **cytometree** seems to do better than Flame on this example compared to manual gating.

cytometree is very fast and leads to population labels similar to those defined by experimentalists. This makes **cytometree** a very practical tool for experimentalists. In addition to being able to estimate proportions of pre-specified subpopulations, it can also be used in a fully unsupervised manner to perform exhaustive gating. It is fully automated and relies on a single tuning parameter, t^* . We performed a sensitivity analysis to show that **cytometree** is robust to the choice of t^* ; the default value of 0.1 worked well in all the samples we have tested (115 for Flowcap I and 60 from HIPC). For these reasons, **cytometree** is likely to play an important role in both clinical and discovery-based research activities.

Gating in **cytometree** is basically done through recursive thresholding of marginal densities based on the assumption that cells express or do not express certain markers, leading to bimodality. This assumption is reasonable in most scientific applications, but some markers (e.g. functional markers) might not be truly bimodal. In this case, these markers would likely not be thresholded and thus would not be represented in the gating tree. Different cases may occur, e.g. a marker may exhibit trimodality: such a feature may be retrieved through the annotation process of **cytometree**, as shown in Figure S1 of the Supplementary Material. A marker may be truly “continuous” and not useful for distinguishing subpopulations. Furthermore, the leaves (or any node) of the tree could then be extracted and further analyzed using other methods, including methods that have been developed to model functional markers (Lin *et al.*, 2015a,b). The populations found could be further annotated using semantic labeling such as

that implemented in flowCL (Courtot *et al.*, 2014). Finally, it should be noted that because of the bimodality assumption **cytometree** is not adapted to gating light scatter channels (i.e. FSC and SSC) and as such it should be applied once these channels have been gated (e.g. applied to the lymphocyte population). The light scatter gates can easily be obtained by importing manual gates using the flowWorkspace package or using algorithms that have been designed to gate these two parameters (e.g. the lymphGate in the flowStats package).

As with all unsupervised algorithms, **cytometree** has difficulties in reliably identifying small populations. For instance, it correctly identified activated T cells in some samples, as shown in Table 1, but failed to identify these small populations in other samples. This result is expected as **cytometree** relies on bimodal marginal distributions to define populations. Moreover, the D criterion is a statistic and as such has a variance that may be large if the number of cells is small; we recommend to stop the search for population sizes lower than 50. We have done a robustness analysis showing that **cytometree** performs well in moderately small samples: results obtained on one fourth of an original sample are very similar to those obtained on the whole sample as shown in Table 1 of Supplementary material. For rare populations, marginal density estimates are unlikely to be clearly bimodal. In such cases, some form of a priori knowledge is probably necessary. Linking the data of different samples by alignment (Cron *et al.*, 2013b) or through the use of random effects as proposed by Pyne *et al.* (2014) could give more stability for rare populations.

In conclusion, the proposed algorithm is very promising in terms of its performance and its computational efficiency, both of which are important considering the pace at which the numbers of markers on single cells that can be measured is increasing.

Funding

This work was supported by the Investissements d’Avenir program managed by the ANR under reference ANR-10-LABX-77.

References

- Aghaeepour, N. *et al.* (2011). Rapid cell population identification in flow cytometry data. *Cytometry Part A*, **79**(1), 6–13.
- Aghaeepour, N. *et al.* (2013). Critical assessment of automated flow cytometry data analysis techniques. *Nature methods*, **10**(3), 228–38.
- Anchang, B. *et al.* (2014). Ccast: a model-based gating strategy to isolate homogeneous subpopulations in a heterogeneous population of single cells. *PLoS computational biology*, **10**(7), e1003664.
- Brinkman, R. R. *et al.* (2015). State-of-the-art in the computational analysis of cytometry data. *Cytometry Part A*, **87**(7), 591–593.
- Brinkman, R. R. *et al.* (2016). Automated analysis of flow cytometry data comes of age. *Cytometry Part A*, **89**(1), 13–15.
- Brusic, V. *et al.* (2014). Computational resources for high-dimensional immune analysis from the human immunology project consortium. *Nat. Biotechnol.*, **32**(2), 146–148.
- Chan, C. *et al.* (2008). Statistical mixture modeling for cell subtype identification in flow cytometry. *Cytometry. Part A : the journal of the International Society for Analytical Cytology*, **73**(8), 693–701.
- Commenges, D. *et al.* (2008). Estimating a difference of kullback–leibler risks using a normalized difference of aic. *The Annals of Applied Statistics*, **2**(3), 1123–1142.
- Courtot, M. *et al.* (2014). flowcl: ontology-based cell population labelling in flow cytometry. *Bioinformatics*, **31**(8), 1337–1339.
- Cron, A. *et al.* (2013a). Hierarchical modeling for rare event detection and cell subset alignment across flow cytometry samples. *PLoS Comput Biol*, **9**(7), e1003130.
- Cron, A. *et al.* (2013b). Hierarchical modeling for rare event detection and cell subset alignment across flow cytometry samples. *PLoS computational biology*, **9**(7), e1003130.
- Finak, G. *et al.* (2009). Merging mixture components for cell population identification in flow cytometry. *Adv. Bioinformatics*, **2009**(2), 247646.
- Finak, G. *et al.* (2014). OpenCyto: An Open Source Infrastructure for Scalable, Robust, Reproducible, and Automated, End-to-End Flow Cytometry Data Analysis. *PLoS Computational Biology*, **10**(8), e1003806.
- Finak, G. *et al.* (2016). Standardizing Flow Cytometry Immunophenotyping Analysis from the Human ImmunoPhenotyping Consortium. *Scientific reports*, **6**(August 2015), 20686.

- Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, **97**(458), 611–631.
- Fraley, C. *et al.* (2012). Normal mixture modeling for model-based clustering, classification, and density estimation. *Department of Statistics, University of Washington*, **23**, 2012.
- Lee, H.-C. *et al.* (2017). Automated cell type discovery and classification through knowledge transfer. *Bioinformatics (Oxford, England)*.
- Li, H. *et al.* (2017). Gating mass cytometry data by deep learning. *Bioinformatics*, **33**(21), 3423–3430.
- Lin, L. *et al.* (2015a). COMPASS identifies T-cell subsets correlated with clinical outcomes. *Nat. Biotechnol.*, **33**(6), 610–616.
- Lin, L. *et al.* (2015b). Identification and visualization of multidimensional antigen-specific T-cell populations in polychromatic cytometry data. *Cytometry A*, **87**(7), 675–682.
- Lo, K. *et al.* (2008). Automated gating of flow cytometry data via robust model-based clustering. *Cytometry. Part A : the journal of the International Society for Analytical Cytology*, **73**(4), 321–32.
- Lux, M. *et al.* (2018). flowlearn: Fast and precise identification and quality checking of cell populations in flow cytometry. *Bioinformatics*.
- Malek, M. *et al.* (2015). FlowDensity: Reproducing manual gating of flow cytometry data by automated density-based cell population identification. *Bioinformatics*, **31**(4), 606–607.
- Naumann, U. and Wand, M. (2009). Automation in high-content flow cytometry screening. *Cytometry Part A*, **75**(9), 789–797.
- O’Neill, K. *et al.* (2013). Flow cytometry bioinformatics. *PLoS Comput Biol*, **9**(12), e1003365.
- Parel, Y. and Chizzolini, C. (2004). CD4+ CD8+ double positive (dp) t cells in health and disease. *Autoimmunity reviews*, **3**(3), 215–220.
- Pyne, S. *et al.* (2009). Automated high-dimensional flow cytometric data analysis. *Proceedings of the National Academy of Sciences of the United States of America*, **106**(21), 8519–24.
- Pyne, S. *et al.* (2014). Joint modeling and registration of cell populations in cohorts of high-dimensional flow cytometric data. *PloS one*, **9**(7), e100334.

- Qian, Y. *et al.* (2010). Elucidation of seventeen human peripheral blood B-cell subsets and quantification of the tetanus response using a density-based method for the automated identification of cell populations in multidimensional flow cytometry data. *Cytometry. Part B, Clinical cytometry*, **78 Suppl 1**(May), S69–82.
- Quandt, D. *et al.* (2014). Peripheral CD4CD8 double positive t cells with a distinct helper cytokine profile are increased in rheumatoid arthritis. *PloS one*, **9**(3), e93293.
- Saeyns, Y. *et al.* (2016). Computational flow cytometry: helping to make sense of high-dimensional immunology data. *Nature Reviews Immunology*, **16**(7), 449–463.
- Samusik, N. *et al.* (2016). Automated mapping of phenotype space with single-cell data. *Nature methods*, **13**(6), 493.
- Zare, H. *et al.* (2009). Data reduction for spectral clustering to analyze high throughput flow cytometry data. *BMC bioinformatics*, **11**, 403–403.
- Zuckermann, F. A. (1999). Extrathymic CD4/CD8 double positive T cells. *Veterinary immunology and immunopathology*, **72**(1), 55–66.

Table 1: T-cell subpopulations evaluated by Stanford for patient 1349 (replicate 3): proportions estimated by manual gating and by `cytometree` are given .

Population Name	Corresponding Markers	cytometree	Stanford
CD4 Activated	CD3+ CD8- CD4+ CD38+ HLADR+	2.55%	1.39%
CD8 Activated	CD3+ CD8+ CD4- CD38+ HLADR+	1.54%	1.67%
CD4 Central Memory	CD3+ CD8- CD4+ CCR7+ CD45RA-	19.86%	19.87%
CD8 Central Memory	CD3+ CD8+ CD4- CCR7+ CD45RA-	2.85%	4%
CD4 Effector	CD3+ CD8- CD4+ CCR7- CD45RA+	1.34%	2.05%
CD8 Effector	CD3+ CD8+ CD4- CCR7- CD45RA+	10.04%	7.94%
CD4 Effector Memory	CD3+ CD8- CD4+ CCR7- CD45RA-	16.23%	18.34%
CD8 Effector Memory	CD3+ CD8+ CD4- CCR7- CD45RA-	8.36%	9.85%
CD4 Naive	CD3+ CD8- CD4+ CCR7+ CD45RA+	13.95%	13.5%
CD8 Naive	CD3+ CD8+ CD4- CCR7+ CD45RA+	22.03%	21.39%

Table 2: F -measures for **cytometree**, with t^* respectively equal to ¹0.05, ²0.1, ³0.15, ⁴0.2, ⁵0.25 and the four algorithms that performed the best on the FlowCAP I challenge data sets. Mean F -measures and mean run times are also given.

Method	GvHD (n=12)	HSCT (n=30)	DLBCL (n=30)	WNV (n=13)	ND (n=30)	Mean	Runtime ^a h:mm:ss ^a
cytometree ¹	0.84	0.90	0.93	0.83	0.88	0.88	00:01:31 [†]
cytometree ²	0.88	0.94	0.93	0.84	0.89	0.90	00:01:24 [†]
cytometree ³	0.92	0.94	0.94	0.88	0.89	0.91	00:01:13 [†]
cytometree ⁴	0.94	0.95	0.93	0.90	0.89	0.92	00:01:06 [†]
cytometree ⁵	0.92	0.95	0.91	0.89	0.89	0.91	00:01:08 [†]
ADICyt	0.81	0.93	0.93	0.86	0.92	0.89	04:50:37
flowMeans	0.88	0.92	0.92	0.88	0.85	0.89	00:02:18
FLAME	0.85	0.94	0.91	0.80	0.90	0.88	00:04:20
Flock	0.84	0.86	0.88	0.83	0.91	0.86	00:00:20

^a Runtime was calculated as time per sample, as displayed in Aghaeepour *et al.* (2013)

[†] Time was extrapolated from flowMeans runtime^a which was ran together with **cytometree** on an Intel(R) i7-4770 CPU @ 3.40 GHz.

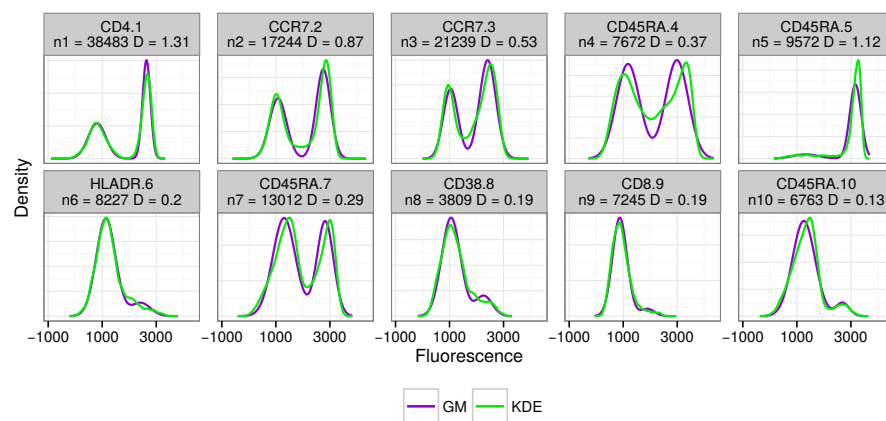


Figure 1: Conditional marginal node distributions for the T cells of patient 1349 in the Stanford FCS data set, replicate 3. In blue, fits obtained with mixtures, in green, non-parametric fits obtained by a kernel method.

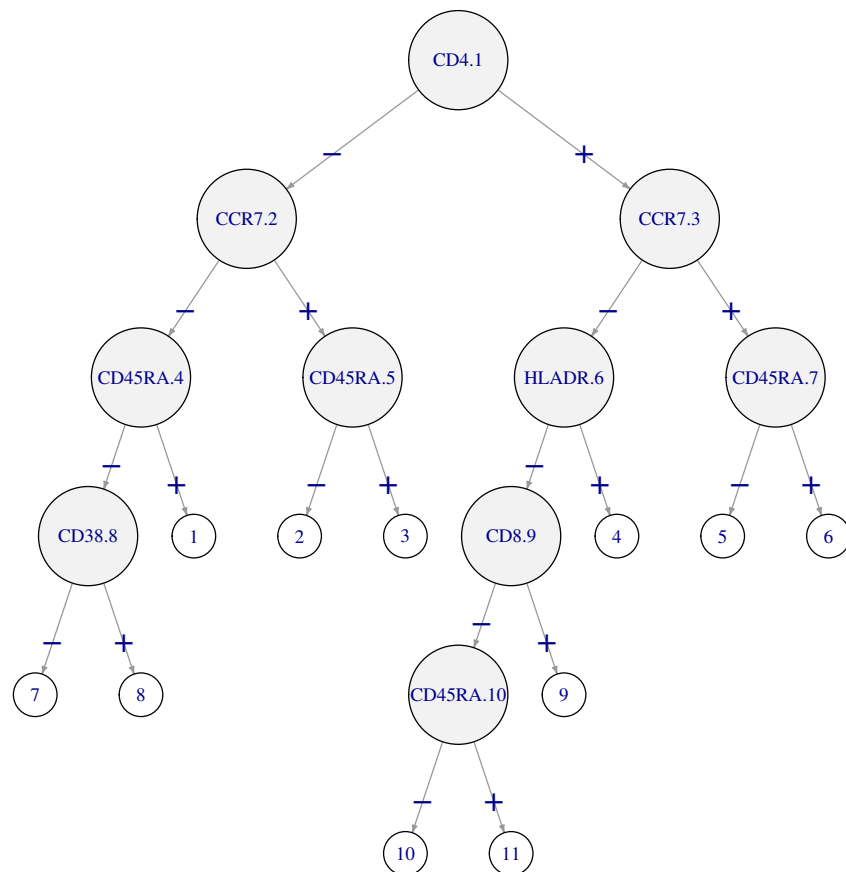


Figure 2: Partitioning tree for the T cells of patient 1349 (replicate 3) from the Stanford dataset. Each node that has children is labeled with the marker on which the subpopulation is split; leaves are numbered.

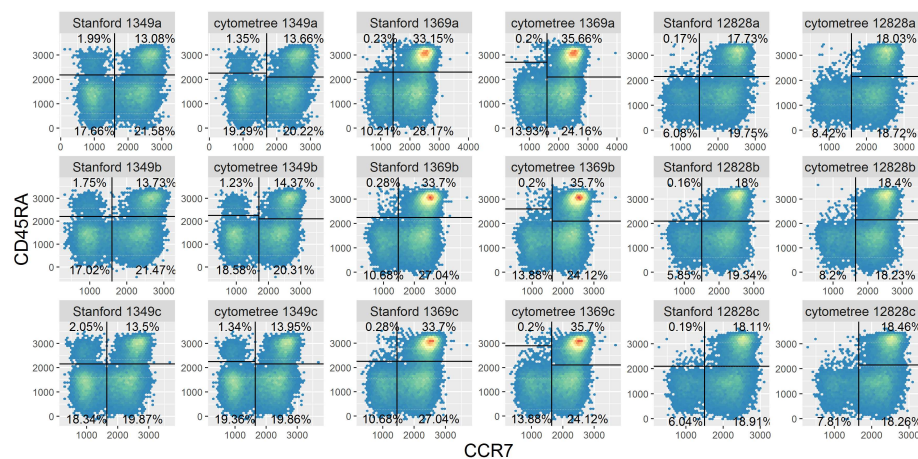


Figure 3: Plots of hexagonally binned data showing four subpopulations of the T-CD4 cells (top left : Effector T cells, top right: Naive T cells: , bottom left : Effector memory T cells , bottom right : Central memory T cells). Gating from Stanford is compared to that obtained by *cytomertree* for the three patients and their three replicates.

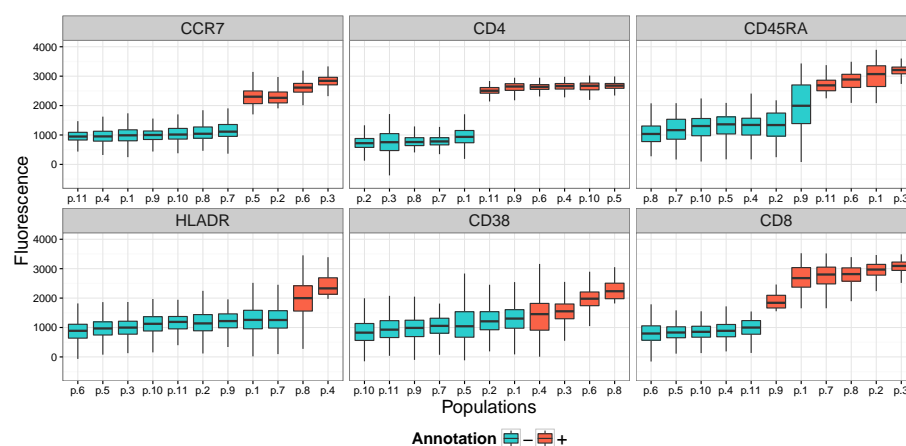


Figure 4: Example of annotation results for the T cells of patient 1349 (replicate c) from the Stanford dataset. Abscissas show the populations from the underlying tree. The clustering algorithm allows allocating the populations into “-” and “+” groups for each marker.

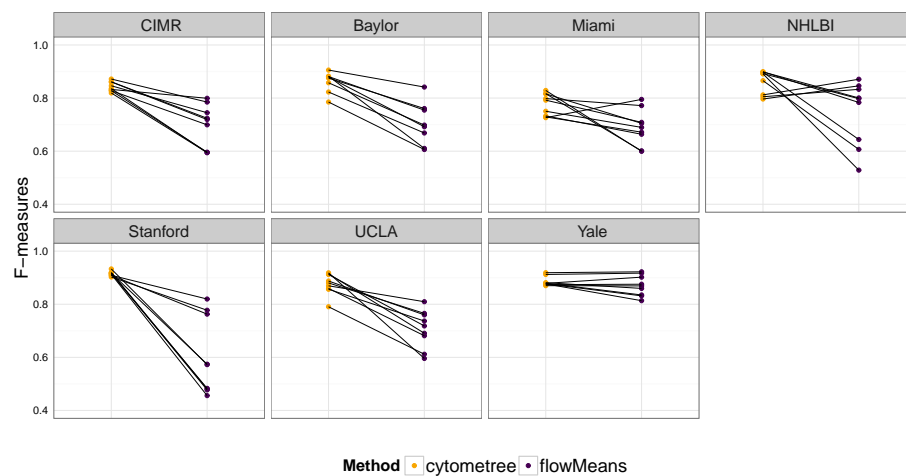


Figure 5: Comparison of F -measures, with respect to manual gating for each of the seven centers, for the nine samples of the T-cell panel of HIPC. Left and blue: **cytometree**; right and red: **flowMeans**.

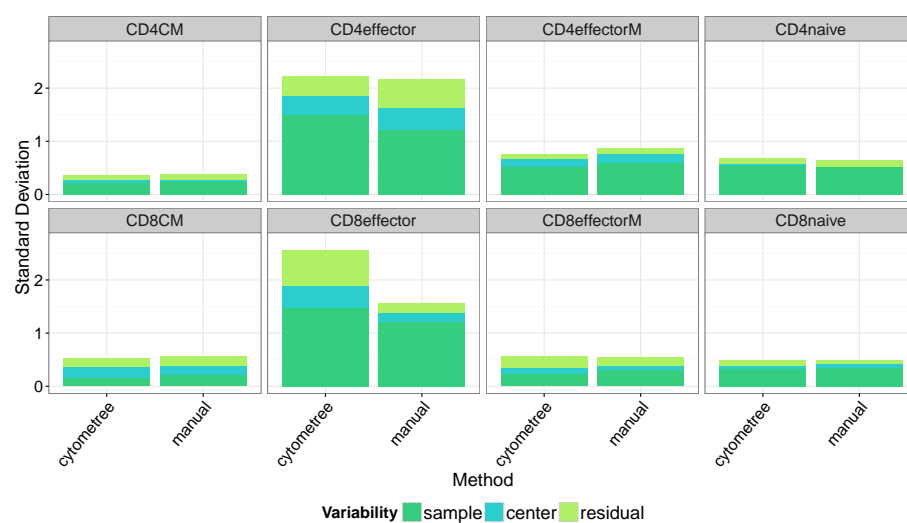


Figure 6: Center, biological, and residual variability for **cytomètre** and the manual gating method for eight subpopulations of the HIPC T-cell panel.